A Model Based Lean Approach to Capability Management

J.P. Venter
CSIR
jpventer@csir.co.za

M. Van Ellewee
CSIR
mvellewee@csir.co.za

ABSTRACT: It is argued that the definition of the required operational capabilities in the short and long term is an essential element of command. Defence Capability Management can be a cumbersome, long and very resource intensive activity. Given the new areas of Defence such as Cyber Warfare, the requirement for agility and within the context of limited resources and tight time constraints a new approach to Defence Capability Management was researched. Ensuring the integration and interaction between long term defence capability planning and short term operational planning is essential in ensuring an effective and efficient means of addressing defence capability management. A process was developed that ensure a close link, by means of using common data constructs based on the Joint Command Control and Collaboration Information Exchange Data Model, between long term capability requirements definition and the short term operational readiness planning. The process requires contextual inputs that are modelled by various means including capability profiles as well as scenario models. It is also essential that the process can provide impactful outputs during the execution of the process, in the absence of complete information, and not only at the end of a process run which could take years. Software applications were developed to support the capability management process and further enhance the efficiency and effective of the process. The approach was tested within landwards operation context and feedback on the results are provided. The conclusion reached was that this approach enables the military stakeholders to engage easier and utilise the capability management approach. The approach also provides a seamless mechanism to support the development of support tools.

1 Introduction

One of the aspects of command is the long and medium planning of forces required for the allocated missions and the short term allocation of forces to specific operations. Especially within the context of limited resources and defense spending in developing countries this is a complex task. Furthermore within the context of a growing cyberwar, that is a fast changing landscape in its own, the planning of required capabilities must be done in an agile manner. This will allow for rapid changes not only in the allocation of capabilities to forces but also on the medium and long term planning as this has a short term impact on the forces that are to be developed.

A substantial amount of the future planning of forces is based on the past experience of long serving defense members. However this experience cannot be relied upon when planning for cyberwar and counter terrorism capabilities as these are fairly new and rapidly changes environments. It is therefore necessary to employ a Capability Management mechanism that can provide answers in the short term, are able to handle continuous changes in the missions and operations, and can deal with uncertainties.

The DODAF/MODAF/UPDM [1]framework enables capability management through the definition of strategic and capabilities views (Hause, 2014). The DODAF/MODAF/UPDM framework identifies a capability viewpoint. Capabilities can be associated with capability configurations that define the systems, organizations and people necessary to achieve the capability (Hause, 2014).

Substantial literature is available on systems design with a capability approach (e.g. Ge et al, 2013). The shortcoming identified was an agile and streamlined approach of linking from a practitioner's strategic and operational planning perspective to the systems design. Another shortcoming identified was the ability to provide interim results and useful information without a complete set of detailed information. As capability management relates to the development of systems of system the following is applicable in this context.

Capability engineering is intended to cope with complexity and uncertainty early in the System-of-Systems (SoS) design process, and to conceptualize the capabilities expected to be achieved by the entire SoS with the continuous evolution of its systems architecture (Levis, 2009). Capability Management as with SoS

[1] DODAF: Department of Defence Architecture Framework, MODAF: Ministry of Defence Architecture Framework, UPDM: Unified Profile for DoDAF/MODAF.

should be aimed toward many satisfactory solutions based on capability requirements to accommodate more possibilities and unpredictable operating environments, as opposed to optimizing the performance of any individual system or an optimal solution to a specific situation (Liles, 2008; Jamshidi, 2009).

Many of the Systems approaches relies on the definition of capability requirements (Under the assumption that the capability requirements of an SoS have been derived via the requirements analysis process to point out the capabilities expected to be achieved, the development of SoS architecture in response to these capability requirements is an iterative architecture modeling and analysis process.)(Ge et al, 2013)

Architectures, frameworks and models have been used in the defense community for more than a decade to address the complexity. However, traditional architecture views are often too complex for decision-makers to readily comprehend. (O'Shea, 2012)) Many of these approaches also rely on the completion of a substantial set of information before useful outputs can be produced.

2 Requirement

A Capability Management approach that is agile, provide the ability to quickly determine the impact and react on changes in the environment, that impact on the design of force capabilities, can accommodate rapid changes in opposition and own force capabilities, and can provide support to defense capability planners in the long, medium and short term is required.

Our experience in using the UPDM framework in our environment is that it requires a substantial set of work from a perspective that was not easily understood by the military participants. The UPDM framework provides a substantial and robust mechanism to address a complete model but does not address the requirement of providing interim answers.

The requirement from our environment was to provide quick feedback utilizing limited resources regarding capabilities required to plan missions and execute operations.

Managing the required capabilities across missions whilst taking into account specific tasks to be executed is complex and beyond the cognitive ability of a single person.

3 The Approach

The process is divided into three major areas shown in Figure 1, namely Mission Planning, Operation Planning and Operation Monitoring. The purpose of clearly indicating these three areas, although the operation planning and operation monitoring may not be seen strictly as part of the capability management effort, is to ensure a consistent data model across all the areas. This ensures that the information is uniform across all the areas. This ensures higher integrity of the data and a consistent language across the areas. This facilitates better communication between the different areas of responsibility.

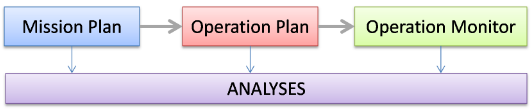


Figure 1: Major Areas of the Capability Management Process.

Mission planning addresses the medium to long term aspects. The elements of the Mission Plan are shown in Figure 2.

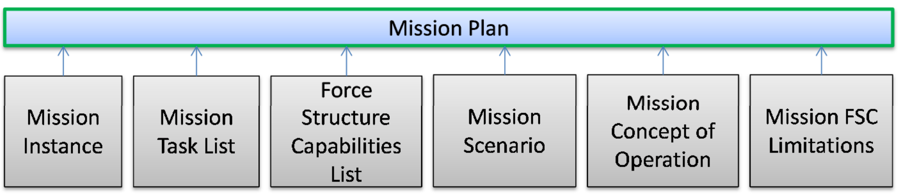


Figure 2: Mission Plan Elements.

Each of the elements in the Mission Plan provides further information that is then used to modify the Mission Task List and the Mission Force Structure Capabilities. Multiple Mission Plans can be developed that would branch into different Mission Plans starting from a broad mission definition. This is shown in Figure 3. Elements used in previous Mission Plans could be reused, as is the case for Scenario 2 in Figure 3. The set of applications developed to support the process further support and enable the re-use.

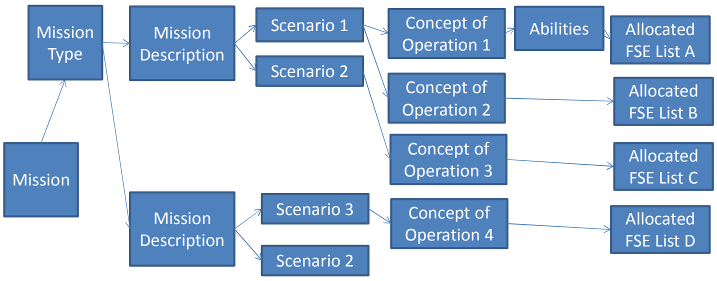


Figure 3: Mission Planning Branches.

At the end a variety of options are available based on well-defined information and defined in a structured and consistent manner. Re-using components enables the development of multiple plans in an agile manner taking into account various possibilities.

Mission planning starts with the definition of a Mission Instance. The Mission Instance captures information regarding a mission. The type of mission (peace keeping, peace enforcement, etc.) is selected from a predefined list of types of missions. The types of mission are based on the mission type definition as used in the JC3IEDM[2] (MIP, 2012). Further information captured in the Mission Instance includes the mission objective, operational concept, required effects, and end state. All of this information is captured in an unstructured format. This enables using information from forums that may not be in a position to use formal structured approaches because the participants may not understand the complexity. Certain elements in the Mission Instance that are initially captured in an unstructured manner will be captured in a more structured manner later in the process.

The next step is to associate tasks with a Mission Instance as part of the Mission Plan. The tasks are selected from a predefine list of tasks. The task list is based on the task list as used in the JC3IDEM but was customized for the South African context. A task list indicates task specifically included but also indicates tasks explicitly excluded. Being able to explicitly exclude tasks provide a mechanism to incorporate mandate or agreement elements in terms of operational approach into the process. In order to facilitate fast tracking the process a set of templates for specific mission types was developed. These templates provide a set of tasks that would normally be associated with a specific mission type without taking into account specific operational concepts or operational conditions. It is therefore more an approach of excluding tasks not applicable to a mission type than including tasks. The templates provide a good starting point and do capture, to some extent, experience from previous missions into the appropriate list of tasks. The task list can be modified based on the specific mission context. The same set of tasks is used but if a task is included or excluded is modified. This modification can be based on, for instance, a specific scenario element that would exclude a specific task.

Defining the required Force Structure Capabilities (FSC) is the next step in the process. At this stage the elements are defined in the form of a generic force structure capability and would not be related to a specific force structure element. The reference would be to an air-to-air fighter capability rather than a specific aircraft element. As a first iteration the FSC is only included or excluded from the Mission Plan. A further refinement is to indicate the role that the FSC play in the mission. The following classification is used for this purpose:

- c = the FSC can / should command (directly or indirectly, taking tactical responsibility, formulating policy for, determining the doctrine of, planning, doing the Joint Interdepartmental Multinational liaison for, etc.) this specific task.
- p = the FSC can / should participate (directly or indirectly, alone, grouped, etc.) in this specific task.
- s = the FSC can / should provide maintenance support (in the sense of logistics, personnel, medical, local procurement, etc.) to another support Force Structure Element (FSE) or to a combat FSE as well as the self-support from each FSE's own reserves for its main function during the execution of this specific task.
- s1 = the FSC can / should provide (potential and / or actual) tactical support (in the sense of observation, mobility, intelligence, illumination, signals relay, etc.) to a combat FSE. “s1” is allocated to a combat or support FSE that cannot / should not "p" or "s" but its in-area availability / deployment / potential involvement in the task will / could definitely tactically assist those FSEs who qualified for a "p" for this specific task.
- n = the FSC cannot (should not) be involved in the execution of this specific task due to the fact that its characteristics / primary effects are inapplicable / unsuitable for current / future requirements.

---

[2] JC3IEDM: Joint Command, Control and Consultation Information Exchange Datamodel. MIP: Multilateral Interoperability Programme

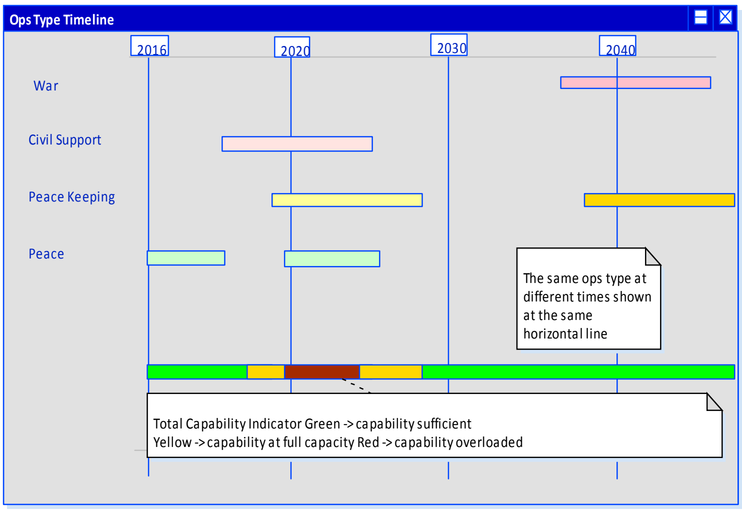
The next element would be to define a Mission Scenario. Currently Mission Scenarios is documented in free text. Research is underway to define a structured model based approach to defining scenarios. Depending on the Mission Scenario the task list and the FSC list can be modified. Each of the modifications to the Mission Task List or Mission FSC list is shown in a separate list. This enables seeing and analyzing the impact that the different elements have on these lists.

Defining Mission Concept of Operation is the next step. As with the scenario the Mission Concept of Operation is currently defined in free text and a structured model approach is being researched. As previously possible changes to the Task List and the FSC List is documented.

The last step in the Mission Plan is to indicate specific FSC Limitations that could influence the planning. This could include the planned phasing out of certain capabilities, or political reasons for not utilizing a specific capability in a mission. These limitations could again have implications on the Mission Task List and FSC List and will be documented as such.

4 Analyses

Due to the consistent and structured definition and capturing of information it is possible to perform various analyses. One example would be to determine the impact on available capabilities based on the time that various mission could be planned for. The overlaps in potential missions analyzed against the available pool of capabilities could provide indications where the required level of capability would not be available.



The design of the Capability Level Indicator screen is shown in Figure 4. The various planned missions are shown on the timeline. Color coding can be used to indicate the priority of a mission. The total capability indicator at the bottom sums the required capability of all the missions that overlap on the timeline. This indicator is color coded. Green indicates that sufficient capability is available to cover all the overlapping missions. Yellow indicates that all capability capacity is utilized to perform the overlapping missions. Red indicates that for the planned missions that overlap on the timeline there would not be sufficient capability.

Figure 4: Capability Level Indicator Design

Further analyses could indicate the tasks that are performed consistently in missions and the associated required capabilities. This could influence the maintenance and upgrading of such capabilities.

One of the aims of the capability management approach is to decrease the level of emotional decisions made. In the absence of specific structured reasoning the requirement for certain capabilities could be argued primarily from an experience and emotional bases. The structured approach enables proving substantiated results rather than loose arguments.

Current force designs could be analyzed against a mission capability plan. For each of the elements in a current force design the task to be performed by that element, as defined in the capability plan, would be assigned. These allocations are then analyzed. This would indicate shortcomings, e.g. a task need to be performed but no capability in the force design can do it, or over design, e.g. a capability is included in the force design that does not contribute to any of the assigned tasks.

5 Future Process Enhancements

Various enhancements can be made to the current process and underlying algorithms can be developed to provide automated inputs to the process. In no way can this process or a system replace the decisions made by the military experts. The purpose is to provide scientific support to the experts in making the final decisions. An example enhancement would be: based on the selected mission type and tasks to suggest a set of FSCs that would be required. This will form a FSC Template similar to the task list template.

6 Supporting the process

Although the defined process is not in essence complex the underlying data and the relationships between the elements are complex. Managing these relationships between the various mission and operational plans requires a well-structured model of the relationships. The rest of this article describes the modelling approach that is supporting the process and software application development.

7 Advantages identified in using a Model Approach in the Capability Management Process

7.1 Visual communication platform

Without using some way of visualizing the Defense Capability Management process would make the development of it very difficult. Capturing the process in a visual platform improves the design by allowing team members to understand and discuss the reasoning for each part of the process and collaborate on it. If the process is visualized it can be showed to the client in order to better explain it and can be used to reassure that the requirements have been met.

A modelling approach was used to visualize and develop the Defense Capability Management process and the supporting elements. Using models makes it easier and more simplistic to understand and explain the process as opposed to writing it in a long document and trying to explain it in words. In Figure 5 an extract from the full capability management process is shown. This diagram shows activities and how they follow on each other (control flows). The diagram also shows information elements (objects) and the flow to and from activities (object flows).

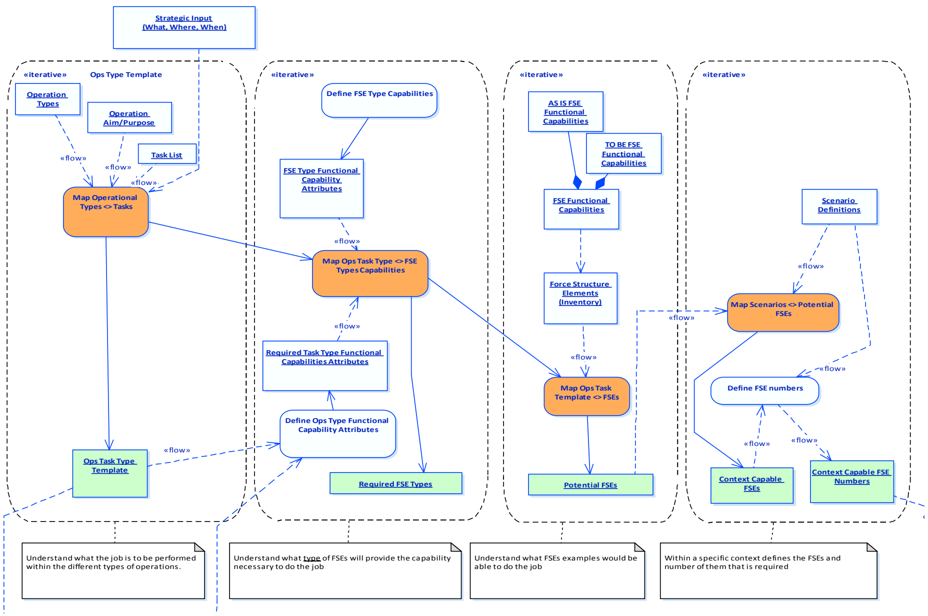


Figure 5: Capability Management Process Extract

An important aspect of the support for the capability management process is the underlying architecture. This architecture defines how elements of the process relate to each other and how the overall structure is composed. The visualization of this architecture is shown in Figure 6.

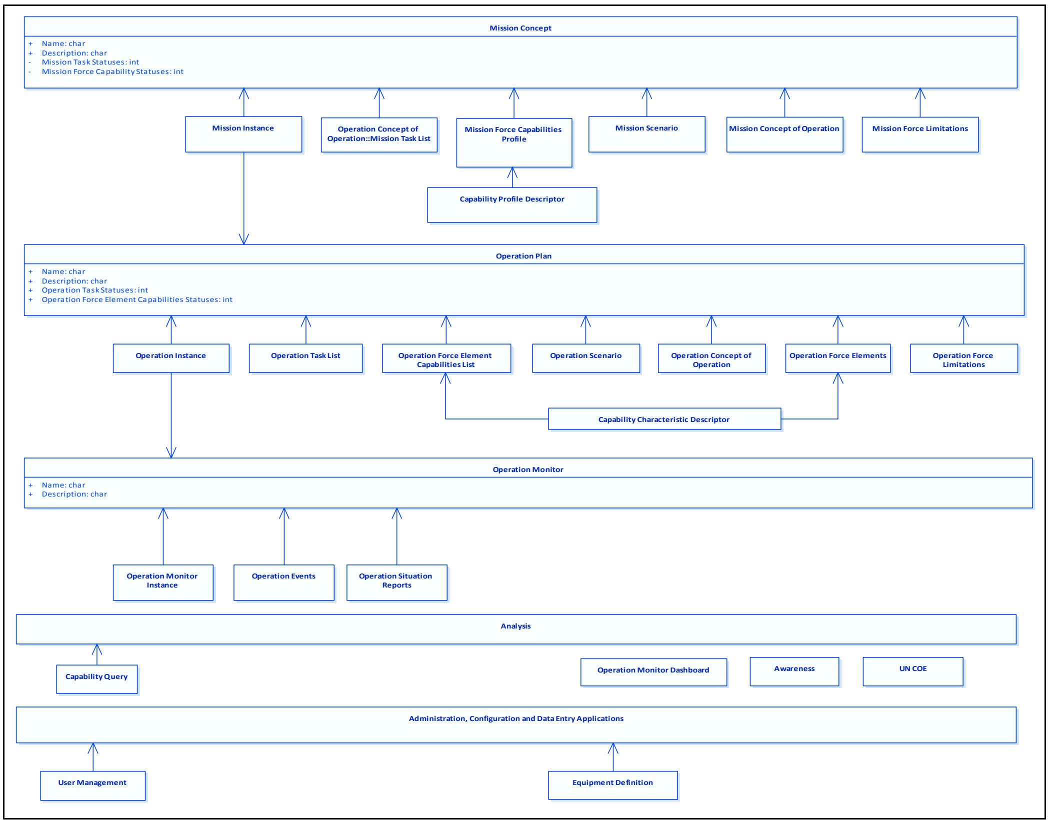


Figure 6: Visualization of Capability Management Architecture

Both of these visualizations utilises basic UML elements and notation. It is therefore quick to explain the notation to all stakeholders and elicit inputs and support.

7.2 Using UML 2 Notation

Since it was decided to use a modelling approach, there are many different modelling notations to consider including Alloy, UML and Z (Jackson, 1999). There are also several different architecture frameworks to consider including the following (Tang et al., 2004):

- Zachman Framework for Enterprise Architecture (ZF)
- Federal Enterprise Architecture Framework (FEAF)
- Open Distributed Processing – Reference Model (RM-ODP)
- The Open Group Architecture Framework (TOGAF)
- Department of Defense Architecture Framework (DoDAF)

Each of these notations and architecture frameworks has their specific strengths and weaknesses (Jackson, 1999, Tang et al., 2004). For example, within the context of the work as described in this paper the DoDAF/MODAF/UPDM architectures are too elaborate and the linking of the MODAF/DODAF/UPDM based definition to the software development to support process requires additional effort.

In developing the Defense Capability Management process limited time and resources were available. Coupled to using one of above mentioned architecture frameworks is the requirement to also use a substantial process. Adequate time and resources were not available to follow such a process.

Thus, due to these constraints the UML notation language was used to model the compact process. The Object Management Group (OMG) has standardized the Unified Modeling Language (UML) and it is now seen as a notation that has emerged as the standard for modelling OO (Object-Oriented) systems (Evans et al., 1998). UML has also since been used as the basis on which some of the above mentioned architecture frameworks are built.

While using the UML notation, it was discovered that the different diagrams and components available are easy to understand making it simplistic to use in comparison to other more complex architecture frameworks. There are only thirteen types of diagrams available in the UML notation which are divided into three categories that is structure diagrams, behavior diagrams and interaction diagrams (OMG, 2016). Of these thirteen diagram types, only 3 diagram types (Use Case Diagrams, Class Diagrams and Activity Diagrams) were used to develop the Defense Capability Management process. This limited amount of diagrams and different categories means less time is required to understand and use the diagrams.

All the models that were produced during the development of the Defense Capability Management process adhered to the semantics and precise UML notation. This made the understanding of the models consistent and reduced the ambiguity in each team members own interpretation of the model.

7.3 Using Sparx System's Enterprise Architect UML modelling tool

A modelling tool is required to create the UML models. Enterprise Architect (Enterprise Architect, 2000) was used as the UML modelling tool for developing the Defense Capability Management process since it is on the list of vendors that are recognized as OMG Members (OMG UML Vendor Directory Listing, 2016). Another reason for using this modelling tool is that is reasonably low-priced when compared to other modelling tools (Blechar and Sinur, 2017).

7.4 Team collaboration

With the limited time constraint to develop the Defense Capability Management process, it was crucial to have a team collaboration platform to enable all team members to view the latest version of the process and work on the project at the same time. Another requirement was for a team member to be able to view and feedback to work in progress. This decrease the development time, decrease rework and resulted in a higher quality output in a shorter time.

In order to meet these team collaboration requirements, the Enterprise Architect project was configured using a Database Management System (DBMS) (Sparx Systems Version Control Best Practices for Enterprise Architect, 2010). This DBMS was installed on a shared server that allowed controlled access to all team members. In setting up this way, all team members could get access to the Enterprise Architect project at the same time. Having the project on a DBMS also allowed for regular backups to be made.

Since several team members had to work on the Defense Capability Management process at the same time, a hierarchy of packages were created to allow different team members to work on different parts of the process. To ensure that only one team member can make changes to a package at a time, Enterprise Architect's User Lock feature was used (Sparx Systems Enterprise Architect User Guide, 2014). With this feature enabled, a team member can only edit a package once it has been locked by that team member. Once done, the team member has to release the user lock allowing other team members to lock it and make changes.

7.5 Traceability

It is essential to have traceability in the Defense Capability Management process to ensure that important information and design decisions doesn't get lost or misinterpreted further down the process (Tang and Han, 2005). Information such as requirements, constraints and process elements are all examples of such important information.

In developing the Defense Capability Management process, requirements, constraints and process elements was easily traced using Enterprise Architect traceability feature (Sparx Systems Enterprise Architect User Guide, 2014). This feature keeps track of every component in the Enterprise Architect as well as any other component it has a relationship with.

The ability to accurately evaluate the impact of changing information like requirements, constraints and process elements is another significant reason why having traceability in the Defense Capability Management process is critical (Tang et al., 2007).

Since Enterprise Architect keep track of all the different relationships a component has to all other components in the project, it is easy to see what other parts in the Defense Capability Management process will be affected if the component is changed (Sparx Systems Enterprise Architect User Guide, 2014).

7.6 Tracking different versions over time

As the Defense Capability Management process evolved over time, different versions were created and changes were made to information like requirements, constraints and process elements. Enterprise Architect not only has the ability to track relationships between different components in a project, but also keep track of these changes made to the components. This is done by using the Baseline feature of Enterprise Architect (Sparx Systems Baseline UML Models, 2010).

The project leader saved a baseline at every minor and major version completed during the development of Defense Capability Management process. With all the different baselines saved, the team members were then able to view any changes between the different versions to better understand and develop on the process with more accuracy and consistency. This also led to a decrease in misinterpretations of the design.

7.7 Document generation

Communicating the process to a variety of clients is essential. This was done by documenting the process by utilising company specific MS Word templates that auto generate certain aspects of the document thus saving time. In order to utilise to the MS Word template, the use of a third party tool was required (eaDocX, 2012).

8 Future Work

The development of algorithms to support the automated population of initial data is research that is planned for the near future. The development of all the separate applications to support the process is in progress and does require substantial future work.

9 Conclusion

A lean Capability Management approach was established by utilizing selected elements from the UML notation. This approach provided sufficient detail to enable the definition of the capability management approach whilst keeping the notation to a small set of elements. The small set of elements reduced the effort of explaining to users how to read and understand the notation. The outcome was that users was more enthusiastic about utilizing the model and contributing to the further definition of the model.

Utilizing the UML notation had the further benefit that providing the necessary inputs for the development of support tools was seamless. It was not necessary to translate from a more elaborate model notation into UML before the software model definition was done. The capability management model elements was used directly to define the requirements for the software design. The overall capability management architecture as shown in Figure 6 was also utilized as the overall approach for the definition of the required software components. The data model definition done for the capability management approach was used to directly define the database design in the UML model which was then used to generate the required database tables automatically.

Overall the approach was shown to be easier to understand by the stakeholders and also provided a seamless integration with the development of the software support tools.

10 References

Blechar, M.J. and Sinur, J. (2007). Magic Quadrant for Business Process Analysis Tools. [pdf] Gartner RAS Core Research Note G, 148777. Available at: http://www.secure.com.sg/courses/ICT353/Session_Collateral/TOP_05_SEM_07_RESEARCH_GARTNER_Magic_Quadrant_BPA_2008.pdf [Accessed 7 Jun. 2017].

eaDocX. (2012) United Kingdom: Ability Engineering Ltd. Available at: http://www.eadocx.com/

Enterprise Architect. (2000) Creswick, Victoria, Australia: Sparx Systems. Available at: http://www.sparxsystems.com/

Evans, A., France, R., Lano, K. and Rumpe, B. (1998). Developing the UML as a formal modelling notation. [pdf] In Proc. UML'98, LNCS (Vol. 1618). Available at: https://wwwbroy.in.tum.de/publ/papers/EFLR98b.pdf [Accessed 7 Jun. 2017].

Ge, B., Hipel, K.W., Yang, K., Chen, Y. , A Data-Centric Capability-Focused Approach for System-of-Systems Architecture Modeling and Analysis, Systems Engineering Journal, Volume 16, Issue 3 Autumn (Fall) 2013 Pages 363–377

Hause, M., The Unified Profile for DoDAF/MODAF (UPDM) Enabling Systems of Systems on Many Levels, Systems Conference, 2010 4th Annual IEEE

Jackson, D. (1999). A comparison of object modelling notations: Alloy, UML and Z. [pdf] MIT Lab for Computer Science. Unpublished manuscript. Available at: http://asa.iti.kit.edu/downloads/alloy-comparison.pdf [Accessed 7 Jun. 2017].

MULTILATERAL INTEROPERABILITY PROGRAMME (MIP), THE JOINT C3 INFORMATION EXCHANGE DATA MODEL (JC3IEDM Main), 2012

O'Shea, Kevin ; Pong, Peter ; Bulluss, Gary, Fit-for-Purpose Visualisation of Architecture to support Defence Capability Decision-Making, DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION (AUSTRALIA) JOINT OPERATIONS DIVISION, Technical note, Jul 2012

Sparx Systems. (2010). Baseline UML Models. [pdf] www.sparxsystems.com. Available at: http://www.sparxsystems.com/downloads/resources/booklets/baseline_uml_models.pdf [Accessed 7 Jun. 2017].

Sparx Systems. (2010). Version Control Best Practices for Enterprise Architect. [pdf] www.sparxsystems.com. Available at: http://www.sparxsystems.com/WhitePapers/Version_Control.pdf [Accessed 7 Jun. 2017].

Sparx Systems. (2014). Enterprise Architect User Guide. [pdf] www.sparxsystems.com. Available at: http://www.sparxsystems.com/bin/EAUserGuide.pdf [Accessed 7 Jun. 2017].

Tang, A. and Han, J. (2005). Architecture rationalization: a methodology for architecture verifiability, traceability and completeness. In Engineering of Computer-Based Systems, 2005. ECBS'05. 12th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (pp. 135-144). IEEE. Available at: http://ieeexplore.ieee.org/document/1409911/#full-text-section [Accessed 5 Jun. 2017].

Tang, A., Jin, Y. and Han, J. (2007). A rationale-based architecture model for design traceability and reasoning. Journal of Systems and Software, [online] Volume 80(6), p. 918-934. Available at: http://www.sciencedirect.com/science/article/pii/S0164121206002287#bib40 [Accessed 7 Jun. 2017].

Tang, A., Han, J. and Chen, P. (2004). A comparative analysis of architecture frameworks. In Software Engineering Conference, 2004. 11th Asia-Pacific (pp. 640-647). IEEE. Available at: http://ieeexplore.ieee.org/document/1371981/#full-text-section [Accessed 7 Jun. 2017].

www.omg.org (2016). Introduction To OMG's Unified Modeling Language® (UML®). [online] Available at: http://www.omg.org/gettingstarted/what_is_uml.htm [Accessed 6 Jun. 2017].

www.omg.org, (2016). UML Vendor Directory Listing. [online] Available at: http://uml-directory.omg.org/vendor/list.htm [Accessed 6 Jun. 2017].